Q1. (10 points): Did you get the same answer as I did? Many of you will. The random number generator is not really random! How could you explain this?

```matlab
>> dropi(10)
ans =
Columns 1 through 7
    0.9501    0.2311    0.6068    0.4860    0.8913    0.7621    0.4565
Columns 8 through 10
    0.0185    0.8214    0.4447
```
Note that the numbers obtained match exactly that in the handout. The `rand` function is a pseudo-RNG, and upon startup the seed is reset to produce the same "random" sequence of numbers. You can change the seed by `rand('state',sum(100*clock))`.

Q2. (5 points): What does `cumsum` do?

```matlab
>> pvec=[.2 .3 .4 .1]
pvec = 0.2000 0.3000 0.4000 0.1000
>> cumsum(pvec)
ans = 0.2000 0.5000 0.9000 1.0000
```
Evidently `cumsum` inputs a vector of numbers and returns a vector of the same length with entry j replaced by the sum of the first j entries in pvec.

Q3. (10 points): Create a sequence of length 2000 and look at its last letter, repeat this, say 50 times. What do you observe, what do you guess is the stationary distribution of the Markov chain defined by the matrix A?

A reasonable estimate of the stationary distribution is the relative frequencies of the last base in each of the 50 simulated chains. Results may vary, but should be close to the true stationary distribution. This can be calculated as the normalized eigenvector corresponding to the unit eigenvalue: A: 0.3636, C: 0.1653, G: 0.1818, T: 0.2893.

Remark: It is not correct to estimate the stationary distribution using the entire sequences of 2000 bases.

Q4. (5 points): Type `log2(CGI./NCG)`.

The output matrix is the score matrix, $\beta$. It contains the log-likelihood ratios of the corresponding transition probabilities. As the logarithm is base 2, the matrix entries are represented in bits.

Q5. (20 points): (This is the real problem, you will need to make up some little auxiliary functions) Compute the 20 scores and make a histogram for them both.

The basic idea: The score of a sequence is the log likelihood ratio of observing this sequence under the CpG island model compared with observing this sequence under the non-CpG island model. For a sequence $x$ of length $L$ ($x_1, x_2, ..., x_L$), the score can be computed by

$$S(x) = \sum_{i=2}^{L} \beta_{x_{i-1}x_i}.$$
function score=score_cal(chain, betamat)
score=0;
for i=2:length(chain)
    score=score+(betamat(chain(i-1), chain(i))); end

Histograms may vary, but one should expect the scores generated by the CpG island model to be mostly positive whereas those generated by the non-CpG island model to be mostly negative. Increased chain length will make violation of those expectations more unlikely.

Remarks:

(a) The genmarkov function in the handout is missing a line: you need to define punif before using it in line 3. Each matlab function, defined by an M-file, has its own local variables, which are separate from those of other functions, and from those of the base workspace and non-function scripts.

(b) In the above algorithm, the first base is only used to compute the score for the second base. Since the first base is simulated under a uniform distribution in both the CpG and non-CpG island model, the first base itself does not contain any information.

Q6. (Bonus, 10 points) We made a simplification in the generation of the Markov chains which was not completely correct, what was it and how can we improve?

The initial state (first base in the simulated chain) was chosen from a uniform prior probability distribution under both the CpG island model and the non-CpG island model. This is not completely correct. A better way is to simulate the first base from the stationary distributions of the CGI and the NCG matrices.

Remarks:

(a) To get the stationary distributions of the transition matrix, we need to compute the eigenvectors CGI’ (NCG’), and not CGI (respectively, NCG).

>> [e v]=eig(CGI')
>> (e(:,1)/sum(e(:,1)))'
an = 0.1543 0.3425 0.3500 0.1532

(b) If the first base is sampled from the stationary distribution, it also contains information that discriminates the two models. A modified score is

\[ S(x) = \log_2 \frac{P(x_1 \mid CGI)}{P(x_1 \mid NCG)} + \sum_{i=2}^{L} \beta x_{i-1} x_i, \]

where \( P(x_1 \mid CGI) \) (\( P(x_1 \mid NCG) \)) is taken from the stationary distribution of CGI (respectively, NCG).

Comments: This assignment was graded out of 100 points: 50 points for trying out matlab and implementing necessary auxillary functions, and 50 points for answering the questions. In the future, please make sure to include only the relevant computer output. For example, if your loop repeatedly outputs

pvec = 0.2000 0.3000 0.4000 0.1000

please remove these lines from your submission. On the other hand, when presenting a figure, please try to include enough information so that a reader can understand what you did: for the histogram in Question 5, you should indicate whether it is a histogram of score or normalized score (score/L) and how long the simulated sequences are.